THE EFFECT OF PLUGGED AND UNPLUGGED APPROACH ON THE DEVELOPMENT OF COMPUTATIONAL THINKING IN SIXTH GRADE STUDENTS

A

Synopsis

Submitted in the partial fulfilment for the requirements of

Doctor of Philosophy

In

Education

Submitted by

Meghna Singh

Under the supervision of

Prof. Archana Kapoor

Co-Supervisor
Professor Emeritus
Faculty of Education

Dr. Arti Singh

Supervisor
Assistant Professor
Dept. of Pedagogical Sciences

DEPARTMENT OF PEDAGOGICAL SCIENCES, FACULTY OF EDUCATION

DAYALBAGH EDUCATIONAL INSTITUTE (DEEMED UNIVERSITY)

DAYALBAGH, AGRA -282005

2021

INTRODUCTION

The technology in this century is developing at a faster rate, and it is going to further transform in the years to come. The technology like AI is estimated to automate, around 40% of jobs that exist today. But, in addition to that, it will open avenues for jobs that require skills like Computational Thinking (hereafter referred to as CT). It was Seymour Papert, who first used the term CT in his book "Mindstorms: Children, Computers and Powerful Ideas" in 1980. He referred CT mainly as a relationship between programming and thinking skills. He was of the opinion that, students' constructions, undertaken while programming with Logo, could boost their procedural thinking across many disciplines. Jeanette Wing is the researcher who revived the interest of researchers in CT. In 2006, she defined it as "solving problems, designing systems, and understanding human behaviour, by drawing on the concepts fundamental to computer science." She also argued that CT, "just like reading, writing, and arithmetic, should be added to every child's analytical ability." The new National Policy on Education rolled out in July 2020, also recognizes this fact, that Computational Thinking should be given emphasis right from the foundational stage of school curriculum.

An operational concept that serves as a significant reference point was created by the Computer Science Teachers Association and the International Society for technology in education (CSTA & ISTE, 2009). This encompasses all of the processes that constitute CT as a practice: "Computational Thinking (CT) is a problem-solving process that includes (but is not limited to) the following characteristics:

• Formulating problems in a way that enables us to use a computer and other tools to help solve them;

• Logical organization and analysis of data;

• Representing data through abstractions such as models and simulations;

• Automating solutions through algorithmic thinking (a series of ordered steps);

• Identifying, analysing, and implementing possible solutions with the goal of achieving the most efficient and effective combination of steps and resources;

• Generalizing and transferring this problem-solving process to a wide variety of problems."

As per Google for Education, Computational Thinking is described as a process which incorporates four techniques, namely decomposition, pattern recognition, generalization and abstraction, and

algorithm design. Computational thinking includes the skills and ways of thinking that are used when writing computer programs, but go beyond the use of computers (Google, 2015). Royal Society (2012) states that "Computational Thinking is the process of recognizing aspects of computation in the world that surrounds us and applying tools and techniques from Computer Science to understand and reason about both natural and artificial systems and processes."

In the August of 2016, the CSTA released CSTA K–12 Computer Science Standards. This update to existing CSTA standards refers to Wing's (2011) CT definitions and stresses the problem-solving aspects, as well as abstraction, automation, and analysis as distinctive elements of CT: "We believe that computational thinking is a problem-solving methodology that expands the realm of computer science into all disciplines, providing a distinct means of analysing and developing solutions to problems that can be solved computationally. With its focus on abstraction, automation, and analysis, CT is a core element of the broader discipline of Computer Science."

Computational thinking's nature includes breakdown of complex problems into more common familiar problems (decomposition of problem), using a series of steps (algorithms) to solve problems, analysing how the solution applies to similar problems (abstraction), and eventually deciding whether a computer can help to resolve those problems more effectively (automation). These steps are fundamental to computer science, but their strength and usefulness extend beyond any one discipline (Yadav, Hong, & Stephenson,2016). Many researchers have tried to come up with their version of the definition of Computational thinking over the years. In this context the core components as described by Yadav et. al (2016) helps us to understand CT better.

Mishra and Yadav (2013) said, "Computational thinking can foster creativity by allowing students to not only be consumers of technology, but also build tools that can have significant impact on society." Therefore, these skills should be developed in the students in their early years of schooling (Fessakis, Gouli, & Mavroudi,2013; Sullivan & Bers,2016).

After establishing the importance of the development of Computational thinking in young students, it becomes important to find out, as to what approaches have been recently employed to ensure its development and to test their effectiveness in the current educational settings. The review of literature

on the ways to develop CT in the students, gives an account of two widely used approaches, i.e., through Plugged and Unplugged Approach.

Plugged Approach refers to the use of programming activities and tasks within the sphere of Computer Science, to develop Computational thinking. Logo was the first programming language that was used with educational aims at elementary level (Feurzeig, Papert, & Lawler, 2011). Papert (1990) was of the opinion that the use of Logo, would help students to develop their understanding since he considered this programming language as "an instrument designed to help change the way you talk about and think about mathematics and writing and the relationship between them". The next one to gain popularity was Scratch, as it is a block based visual programming language. It is a widely studied language and has shown positive results in improving young people's creativity and reasoning.

Although it is important to mention the existence of a significant amount of research (Bar-On, 1986; Howe, Ross, Johnson, Plane, & Inglis, 1982; Layman & Hall, 1988; Pea, Kurland, & Hawkins, 1985) that shows eventual pedagogical gains derived from the use of Logo, most of the studies were conducted mainly with high school students. Instead, CT skills should be developed at an early age (Buitrago Flórez et al., 2017). After Logo's presence, many programming languages, modelled after it, started to rise with the goal of introducing coding activities into teaching settings in K-12 education.

Among these environments, Scratch stands out as one of the most popular and studied programming languages. It was developed by MIT Media Lab in 2007.It is easy to understand for children as young as eight years old. This visual language is in the shape of blocks, and it allows users to create online projects, games and apps. There are seven concepts that are highly useful in a wide range of Scratch projects, and which transfer to other programming (and non-programming) contexts: sequences, loops, parallelism, events, conditionals, operators, and data.

Sequences: A key concept in programming is that a particular activity or task is expressed as a series of individual steps or instructions that can be executed by the computer.

Loops: They are a mechanism for running the same sequence multiple times.

Events: One thing causing another thing to happen.

Parallelism: Sequences of instructions happening at the same time.

Conditionals: The ability to make decisions based on certain conditions, which supports the expression of multiple outcomes.

Operators: They provide support for mathematical, logical, and string expressions, enabling the programmer to perform numeric and string manipulations.

Data: It involves storing, retrieving, and updating values.

It was created to help young people learn to think creatively, reason out in a systematic way and work collaboratively (Brennan, Balch, & Chung, 2014). Funke, Geldreich, & Hubwieser (2017) did a study on fourth-grade pupils and found out that all of them learned programming with scratch in only three days.

As for the Unplugged Approach, it came as a substitute of Plugged Approach for classrooms lacking the facility of Computers. This requires the teaching of components of Computational thinking, by integrating them with the traditional classroom lesson plans. The idea of 'unplugged' activities was made popular by the team at CS Unplugged, whose website is repository of free learning activities that take place off screen. The CS unplugged activities help to introduce topics in computer science, with no need to learn programming first (Bell & Vahrenhold, 2018). These activities are simpler and can be performed without the use of computer.

Unplugged activities help to carry out interdisciplinary activities without subject boundaries. Students can view the relationship(s) between different subjects in a better way. These activities mostly include physical movements. It is guided by the theories of learning by doing and learning by moving (Bell, Witten, & Fellows, 2015; Rodriguez, Kennicutt, Rader & Camp, 2017; Thies & Vahrenhold, 2013). The activities allow for teamwork and collaboration along participants. Also, they help in the development of communication skill, as the students have to speak about a problem: they formulate the problem, solving strategy, and its solution (Feaster, Segars, Wahba & Hallstrom, 2011; Bell et al., 2015). Researchers active in this field have shown significant effects of unplugged activities on the development of Computational thinking in young students.

EMERGENCE AND JUSTIFICATION OF THE PROBLEM

The ever-changing technology requires the teachers to build in their students, the skills of the 21st century. Computational thinking is one such skill which stands out from the rest, and helps a person to

effectively solve complex problems in any given condition. A lot of European countries have already embedded Computational Thinking skills in their K-12 curriculum. Our country has also realised the importance of introducing CT in the existing curriculum. The National Policy on Education 2020 states, "It is recognized that mathematics and mathematical thinking will be very important for India's future and India's leadership role in the numerous upcoming fields and professions that will involve artificial intelligence, machine learning, and data science, etc. Thus, mathematics and computational thinking will be given increased emphasis throughout the school years, starting with the foundational stage, through a variety of innovative methods, including the regular use of puzzles and games that make mathematical thinking more enjoyable and engaging. Activities involving coding will be introduced in Middle Stage." Keeping this in mind, the researcher wants to take up a study focussing on the development of CT in sixth grade students. As per Piaget's theory and stages of cognitive development, the children of this age group are able to use symbols to relate abstract concepts. They are able to develop working hypotheses and test them. They are able to handle open –ended problems. All these characteristics can allow the teachers to easily initiate the development of CT, without overburdening them with complex activities.

However, the question which still needs to be addressed is how to develop it in the young children in an effective way, so that they can apply it across multiple disciplines. The review of related literature brings into light two approaches to develop Computational thinking in the young students. These two approaches are Plugged Approach and Unplugged Approach. Some of the studies found related to the variables are as follows:

Researches related to Plugged Approach:

Saritapeci (2020) conducted a study on the Development of Computational Thinking Skills of High School Students using Design-Based Learning Activities and Programming Tasks. The study found evidence to suggest that there was a significant positive development in the CT levels after the experimentation in both groups. The most remarkable result to emerge from the data was that DBL activities also had a similar effect on CT skills compared to programming training. Rodriguez-martinez, Gonzalez-calero & Saez-lopez (2019) did research on developing Computational Thinking and Mathematics using Scratch with sixth grade students. The study showed that Scratch can be used

to develop both students' mathematical skills and Computational thinking. Nouri, Zhang, Mannila & Noren (2019) conducted a study on the development of Computational Thinking, digital competence and 21st century skills when learning programming in K-9. In the study 19 Swedish teachers were interviewed who had been teaching programming for a couple of years on their own initiative. The findings indicated that the pupils developed an understanding of fundamental computational concepts. Boucinha, Augusto, Barone, Teresinha, Brackmann & Schneider (2019) performed a study to find out relationship between the Learning of Computational Thinking and the Development of Reasoning of elementary level students. The results showed that, an increase in Computational Thinking favoured the development of Verbal Reasoning, Abstract Reasoning and Mechanical Reasoning. Pinto-Llorente, Martin, Gonzalez & Garcia-Penalvo (2016) did a study to study the effect of the Visual Programming Tool on the Development of Computational Thinking. The Visual Programming Tool was lego education Wedo. The results showed that the programming projects fostered participants' computational thinking.

Researches related to Unplugged Approach:

Bell & Lodi (2019) conducted a study on how to construct Computational Thinking without using Computers. The study employed theoretical approach and the findings suggested that CS activities map well onto commonly accepted CT concepts. Brackmann, Roman-Gonzalez, Robles, Moreno-Leon, Casali & Barone (2017) did research on the development of Computational thinking skills in Spanish primary school, using Unplugged approach. They found positive results, as the students of experimental group outperformed the students of control group. Conde, Fernandez-Llamas, Rodriguez-Sedano, Guerrero-Higueras, Matellan-Olivera, & Garcia-Penalvo (2017) conducted a study regarding the promotion of Computational thinking in K-12 students by applying Unplugged methods and Robotics. The study showed positive outcomes as the students in the experimental group showed significant gain in CT skills as compared to the students in Control group.

The review of related literature shows that there has been much work done on the effect of plugged and unplugged activities in the development of Computational thinking of students at the international level. There have been few researches done at national level on Computational thinking, which does not directly relate to my area of interest. In the studies mentioned above, it was found out that there

were problems with research methodology, like choosing of sample, and delimitations of the study, which can lead to invalid and unreliable results. Furthermore, very few studies were conducted using both plugged and unplugged approaches in the development of Computational Thinking of middle school students. The questions that arise after reviewing the related researches are:

1. Is there any effect of Plugged Approach on the development of Computational thinking in sixth grade students?
2. Is there any effect of Unplugged Approach on the development of Computational thinking in sixth grade students?
3. Which of the two approaches, Plugged or Unplugged, is better for sixth grade students, keeping in mind the Indian classroom situations?

In an attempt to find answer to these questions, the researcher wants to take up the proposed study. Also, this work would be an addition to the existing knowledge of the role of plugged and unplugged approaches in the development of Computational thinking in K-12 students.

STATEMENT OF PROBLEM

The development of Computational thinking in every child, is going to presume an important role in the coming years. This is due to the fact that, Computational thinking has been identified as an important 21st century skill, which will be required by the children to build a promising career in an ever-advancing technological world. Therefore, it is important to find out ways which are suitable to develop it in the young students, keeping in mind the constraints of Indian classrooms situations. In the light of this justification, the problem can be stated as:

The effect of Plugged and Unplugged Approach on the development of Computational Thinking in sixth grade students

DEFINITIONS OF THE TERMS USED IN STUDY

1. Computational Thinking

Computer Science Teachers Association and the International Society for Technology in Education (CSTA & ISTE, 2009) have defined it as: “Computational Thinking (CT) is a problem-solving process that includes (but is not limited to) the following characteristics:

• Formulating problems in a way that enables us to use a computer and other tools to help solve them;

• Logical organization and analysis of data;

• Representing data through abstractions such as models and simulations;

• Automating solutions through algorithmic thinking (a series of ordered steps);

• Identifying, analysing, and implementing possible solutions with the goal of achieving the most efficient and effective combination of steps and resources;

• Generalizing and transferring this problem-solving process to a wide variety of problems"

In the present study, Computational Thinking is a problem-solving thought process that includes elements of problem decomposition, pattern recognition, abstraction and algorithmic thinking.

Problem decomposition refers to breaking down a complex problem or system into smaller, more manageable parts. Pattern recognition means looking for similarities among and within problems. Abstraction is focusing on the important information only, while ignoring any irrelevant detail. Algorithmic thinking refers to developing a step-by-step solution to the problem.

2. Plugged Approach

Plugged approach is a digital approach which involves the use of coding activities, within the sphere of Computer Science, to develop Computational thinking in the students.

3. Unplugged Approach

Unplugged approach is a non-digital approach which requires the integration of Computational thinking elements across various subjects in regular classroom teaching.

OBJECTIVES OF THE STUDY

1. To find out the effect of Plugged Approach on the development of Computational Thinking in sixth grade students.
2. To find out the effect of Unplugged Approach on the development of Computational Thinking in sixth grade students.
3. To compare the effects of Plugged and Unplugged Approach on the development of Computational Thinking in sixth grade students.

HYPOTHESES OF THE STUDY

1. There is significant effect of Plugged Approach on the development of Computational Thinking in sixth grade students.

2. There is significant effect of Unplugged Approach on the development of Computational Thinking in sixth grade students.
3. There is no significant difference between the effects of Plugged Approach and Unplugged Approach on the development of Computational Thinking in sixth grade students.

VARIABLES OF THE STUDY

1. **Independent Variables**: Plugged Approach, Unplugged Approach
2. **Dependent Variable:** Computational Thinking
3. **Extraneous Variables** will be identified and controlled as given in the table below**:**

Table 1: Identification and Controlling of Extraneous Variables

Category of Variables	Extraneous Variables	Controlling Techniques	Methods to control Extraneous Variables
Participant related Variables	Age	Sample selection	Randomization
	Anxiety	Rapport establishment	Psychologically by researcher
	Motivation		
Environment related Variables	Classroom climate	Rapport establishment	Psychologically by researcher

DELIMITATIONS OF THE STUDY

The present study will be delimited to:

- Use of Scratch (programming language) in the Plugged Approach
- Lesson plans in Science for Unplugged approach
- Students of age group 11-13,
- Schools of C.B.S.E. Board of Firozabad district only.

RESEARCH METHOD

The present study will use Quasi-Experimental Method. There will be two groups. One will be experimental group 1, on which the effect of Plugged Approach in the development of Computational Thinking will be observed, and the other will be Experimental group 2, on which the effect of Unplugged Approach in the development of Computational Thinking will be observed.

PROCEDURE OF THE STUDY

The study will take place in three phases: The Pre-intervention Phase, the Intervention Phase and the Post-intervention Phase.

Phase 1: The Pre-intervention Phase (Planning and Developmental Phase)

- In this phase, a Computational thinking tool to measure the Computational thinking of sixth grade students will be developed.
- An intervention program for both, Experimental group 1 and 2, will be developed. This will include modules of Scratch programming language for Experimental group 1(Plugged approach), and 25 lesson plans in science (with Computational thinking elements) for experimental group 2. The learning outcomes given by NCERT for elementary stage does not explicitly mention the Computational thinking elements. However, they are implicitly involved in the learning outcomes given for various subjects, predominantly in Maths and Science.

Table 2: Learning outcomes and to Computational Thinking elements related it

Learning Outcomes given by NCERT for class 6 (Science)	**Computational Thinking elements related to it**
The learner - identifies materials and organisms, such as, plant fibres, flowers, on the basis of observable features, i.e., appearance, texture, function	Pattern recognition
conducts simple investigations to seek answers to queries, e.g., What are the food nutrients present in animal fodder?	Decomposition
constructs models using materials from surroundings and explains their working, e.g., pinhole camera, periscope, electric torch, etc.	Abstraction
draws diagrams/flow charts of organisms and processes, e.g., parts of flower, joints, water cycle, etc.	Algorithmic thinking

- Similarly, other learning outcomes can also be identified while developing the lesson plans in the unplugged approach.
- The initial levels of Computational Thinking of the participants will be measured using a pre-test.

Phase 2: The Intervention Phase (Execution Phase)

- In this phase, an intervention program will be organized for both Experimental group 1, and 2. The intervention period will be of 30 working days.

Phase 3: The Post-intervention Phase (Testing of Hypotheses)

- In this phase, the attained levels of Computational Thinking of the participants will be measured using a post-test.

- The hypotheses will then be tested by comparing the pre-test and post-test results.

RESEARCH DESIGN

The present study will be executed as given in the table below:

Table 3: Research design of the study

Groups	Pre-Intervention Phase	Intervention Phase	Post-Intervention Phase
Experimental 1 (Plugged) [30 participants]	Administration of self-constructed Computational Thinking tool.	The participants will be exposed to scratch programming modules for one month.	Administration of self-constructed Computational Thinking tool.
Experimental 2 (Unplugged) [30 participants]		25 lessons (with Computational thinking elements) will be taught.	

SAMPLE OF THE STUDY

Selection of School: A single CBSE board school will be selected using Purposive Sampling Technique.

Selection of class: Two sections from class 6 will be selected using random sampling.

Selection of participants: A total of 60 students (30 in experimental group 1, and 30 in experimental group 2) will be randomly selected.

RESEARCH TOOLS

A Computational Thinking tool will be developed by the researcher. The tool will be constructed as per general guidelines given below:

(i) Conceptualisation of elements to measure

(ii) Compilation of items and scales

(iii) Testing

(iv) Validation of tool, which includes testing of validity and reliability

STATISTICAL TECHNIQUES

To analyse and interpret the data, both descriptive and inferential statistical techniques will be used.

(a) Descriptive statistical techniques:

1. **Mean**: It will be used to find out the average pre-test and post-test scores of both the experimental groups.

2. **Standard Deviation:** It will be used to find out the variation in the scores of the two groups with respect to their mean values**.**

(b) Inferential statistical techniques:

1. **t-test**: It will be employed to find out statistically significant difference between the mean pre-test and post-test scores of the two experimental groups, as well as between the mean post-test scores of the two experimental groups.

SIGNIFICANCE OF THE STUDY

The proposed study will have a direct influence on the students of middle school level, teachers, curriculum planners, policy makers, teacher educators, educational technologists and the educational researchers. The development of this in-demand skill will prepare the students to take up advanced courses on Artificial intelligence and Machine Learning, and enable them to enter a technology driven future. It will promote higher order thinking skills in the students. It will help the teachers to know about the attitudes and behaviour, regarding the acceptance level of these approaches in actual classroom settings. It will allow teachers to plan out strategies regarding the development of the skill in their students. It will promote policy planners to envision a goal to transform the education to meet the needs of the competitive future. It will guide curriculum planners to provide outline regarding the incorporation of Computational Thinking as a compulsory skill in the existing curriculum. It will help the teacher educators to train pre-service teachers in the developments of this skill in the young students. Also, it may help the in-service teachers to make a choice regarding which approach to use and get informed about their merits and demerits. It will guide educational content developers and educational technologists to build apps, projects, worksheets, and modules with an idea to implicitly impart Computational Thinking skills to their consumers. Also, the proposed study will add to the existing knowledge, concerning the development of Computational Thinking in the young students.

REFERENCES

Bar-On, E. (1986). A programming approach to mathematics. *Computers & Education*, 10(4), 393–401. doi:10.1016/0360- 1315(86)90015-1

Barr, V., & Stephenson, C. (2011). Bringing Computational Thinking to K-12: What is Involved and What is the Role of the Computer Science Education Community? ACM Inroads, 2(1), 48–54.

Barr, D., Harrison, J., & Conery, L. (2011). Computational Thinking: A Digital Age Skill for Everyone. Learning & Leading with Technology, 38(6), 20–23.

Bell, T., & Lodi, M.(2019). Constructing Computational Thinking Without Using Computers. *Constructivist foundations*, Vrije Universiteit Brussel, 2019, Special Issue "Constructionism and Computational Thinking", 14 (3), pp.342-351. ⟨hal-02378761⟩

Bell, T., & Vahrenhold, J.(2018). CS Unplugged-How Is It Used, and Does It Work?. In: Bockenhauer, H.J., Komm, D., Unger, W. (eds) Adventures Between Lower Bounds and Higher Altitudes. Lectures Notes in Computer Science, vol 11011. Springer, Cham. https://doi.org/10.1007/978-3-319-98355-4_29

Boucinha, R.M., Augusto, D., Barone, C., Teresinha, J., Brackmann, C.P., & Schneider, A.M. (2019). Relationship between the Learning of Computational thinking and the Development of Reasoning. *International Journal of Advanced Engineering Research and Science*, *6*(6). Retrieved from http://journal-repository.com/index.php/ijaers/article/view/657

Brackmann, C.P., Roman-Gonzalez, M., Robles, G., Moreno-Leon, J., Casali, A., & Barone, D. (2017). Development of Computational Thinking Skills through Unplugged Activities in Primary School. Proceedings of the 12th Workshop on Primary and Secondary Computing Education, 65-72. https://doi.org/10.1145/3137065.3137069

Brennan, K., Balch, C., & Chung, M.(2014). *Creative computing*. Retrieved from https://scratched.gse.harvard.edu/guide/

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In AERA2012 - annual meeting of the American Educational Research Association. Vancouver, Canada.

Buitrago Flórez, F., Casallas, R., Hernández,M., Reyes, A., Restrepo, S., & Danies, G. (2017). Changing a generation's way of thinking: Teaching computational thinking through programming. *Review of Educational Research*, 87(4), 834–860. doi:10.3102/0034654317710096

Conde, M. A., Fernandez-Llamas, C., Rodriguez-Sedano, F.J., Guerrero-Higueras, A. M., Matellan-Olivera, V., & Garcia-Penalvo, F. J.(2017). Promoting Computational Thinking in K-12

students by applying unplugged methods and robotics. *Proceedings of the 5th Internatioal Conference on Technological Ecosystems for Enhancing Multiculturality*,1-6. https://doi.org/10.1145/3144826.3145355

CSTA & ISTE. (2009). Operational definition of Computational Thinking in K-12 education. Retrieved from https://csta.acm.org/Curriculum/sub/CurrFiles/CompThinkingFlyer.pdf

Feaster, Y., Segars, L., Wahba, S.K., & Hallstrom, J.O.(2011).Teaching CS Unplugged in the High School (with Limited Success). *Proceedings of the 16th Annual Joint Conference on Innovation Technology in Computer Science Education, ITiCSE* '11 (pp. 248–252). Darmstadt, Germany: ACM. doi:10.1145/1999747.1999817

Fessakis, G., Gouli., & Mavroudi, E.(2013). Problem-solving by 5-6 years old kindergarten children in a computer programming environment: A case study. *Computers & Education*, 63, 87-97.

Feurzeig, W., Papert, S. A., & Lawler, B. (2011). Programming-languages as a conceptual framework for teaching mathematics. *Interactive Learning Environments*,19(5), 487-501. doi: 10.1080/10494820903520040

Funke, A., Geldreich, K., & Hubwieser, P.(2017).Analysis of scratch projects of an introductory programming course for primary school students. *Proceedings of the IEEE Global Engineering Education Conference (EDUCON), Athens*, 2017, pp. 1229-1236.

Google for Education.What is Computational Thinking? Computational Thinking for educators, 2015. Retrieved on January 15,2020,fromhttps://computationalthinkingcourse.withgoogle.com/unit? Lesson=8&unit=1.

Gretter, S., & Yadav, A. (2016). Computational Thinking and Media & Information Literacy: An Integrated Approach to Teaching Twenty-First Century Skills. *TechTrends*, 1–7.

Grover, S., & Pea, R. (2013). Computational Thinking in K–12 A Review of the State of the Field. *Educational Researcher*, 42(1), 38–43.

Howe,J. A. M., Ross, P. M., Johnson,K. R., Plane, F., & Inglis, R. (1982). Teaching mathematics through programming in the classroom. *Computers & Education*, 6(1), 85–91. doi:10.1016/0360-1315(82)90016-1

Inhelder, B., & Piaget, J.(1958). *The growth of logical thinking from childhood to adolescence*. New York: Basic Books

Layman, J., & Hall, W. (1988). Logo: A cause for concern. *Computers & Education*, 12(1), 107–112. doi:10.1016/0360-1315 (88)90063-2

Lee, Y. (2020). Quantifying the effects of programming education on students' Knowledge Representation & Perception in Computational Thinking.https://doi.org/10.31219/osf.io/zhetf

Ministry of Human Resource Development. (2020). *National Education Policy*. https://www.education.gov.in/sites/upload_files/mhrd/files/NEP_Final_Englis.pdf

Mishra, P., & Yadav, A. (2013). Of art and algorithm: rethinking technology & creativity in the 21st century. *TechTrends*, 57(3), 10–14. doi:10.1007/s11528-013-0668-7.

National Council of Educational Research and Training. (2017). *Learning outcomes at Elementary Stage*. Ministry of Human Resource Development. https://ncert.nic.in/tilops101.pdf

Nouri, J., Zhang, L., Mannila, L., & Noren, E.(2020). Development of computational thinking , digital competence and 21 century skills when learning programming in K-9. *Education Inquiry*, 11(1), 1-17. https://doi.org/10.1080/20004508.2019.1627844

Papert,S.(1980).*Mindstorms: Children, computers, and powerful ideas.* New York: Basic Books

Pea, R. D., Kurland, D. M., & Hawkins, J. (1985). Logo and the development of thinking skills. In M. Chen & W. Paisley (Eds.), Children and microcomputers: Research on the newest medium (pp. 193–317). Norwood, NJ: Ablex Publishing Corp.

Pinto-Llorente, A. M., Martin, S. C., Gonzalez, M.C, & Garcia-Penalvo, F.J.(2016). Developing computational thinking via the visual programming tool: lego education WeDo. *Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality*, 45-50. https://doi.org/10.1145/3012430.3012495

Rodriguez, B., Kennicutt, S., Rader, C., & Camp, T. (2017). Assessing computational thinking in CS unplugged activities. *Proceedings of the Conference on Integrating Technology into Computer Science Education, ITiCSE*, 501-506.https://doi.org/10.1145/3017680.3017779

Rodriguez-martinez, J. A., Gonzalez-calero, J.A., & Saez-lopez, M.(2019). Computational thinking and mathematics using Scratch: an experiment with sixth-grade students. Interactive Learning Environments, 0(0), 1-12. https://doi.org/10.1080/10494820.2019.1612448

Saritepeci, M.(2020) Developing Computational Thinking Skills of High School Students: Design-Based Learning Activities and Programming Tasks. *Asia-Pacific Education Researcher 29*, 35-54. https://doi.org/10.1007/s40299-019-00480-2

Selby, C. C. (2014). How Can the Teaching of Programming Be Used to Enhance Computational Thinking Skills? (2). University of Southampthon, Southampthon, UK.

Sullivan, A., & Bers, M. U. (2016). Robotics in the early childhood classroom: Learning outcomes from an 8-week robotics curriculum in pre-kindergarten through second grade. *International Journal of Technology and Design Education*, 26(1), 3–20

Thies, R., Vahrenhold, J. (2013). On plugging "unplugged" into CS classes. In: Special Interest Group on Computer Science Education, Denver, 2013.

Wing, J. M. (2006). Computational thinking. Communications of the ACM, 49(3), 33–35.

Wing, J. M. (2008). Computational thinking and thinking about computing. Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, 366(1881), 3717–3725.

Yadav,A., Krist,C.,Good,J., & Caeli,E.N. (2018) Computational thinking in elementary classrooms: measuring teacher understanding of computational ideas for teaching science, Computer Science Education, 28:4, 371-400, DOI: 10.1080/08993408.2018.1560550